



FILAS

Allan Robson



DCA0204 – Estruturas de Dados

## Estruturas de Dados

- Uma **Estrutura de Dados** é um modo particular de **armazenamento e organização de dados** em um computador de modo que possam ser usados eficientemente, **facilitando sua busca e modificação**.
- Exemplos:
  - **Pilhas;**
  - **Filas;**
  - Árvores (Árvore binária, Rubro-Negra, AVL, ...);
  - Tabelas de Dispersão (*Hash Table*)
  - ...

Fila

- Suponha que devemos resolver o problema de **organizar as pessoas que querem ser atendidas num guichê.**



- Suponha que devemos resolver o problema de **organizar as pessoas que querem ser atendidas num guichê**.
- Nesse problema, algumas regras devem ser atendidas:
  - A primeira **pessoa a ser atendida será a primeira que entrou na fila**;
  - Sempre que uma nova pessoa deseja atendimento, ela deve **entrar no final da fila**;
  - Não é permitido “furar” a fila, ou seja, entrar uma pessoa entre outras que já estão presentes.

## ➡ Definição

- Filas são listas lineares que adotam a política FIFO (*First In First Out*) – o primeiro que entra é o primeiro que sai;
- As inserções de novos elementos devem ser feitas sempre o final da fila;
- Apenas o elemento que está no início da fila é que pode ser removido;

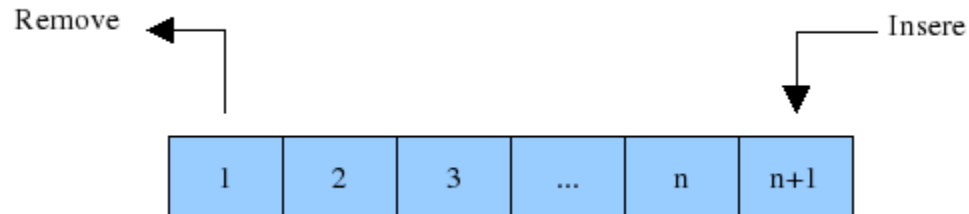
- Aplicações:
  - Alocação de recursos para impressão de documentos (*spooler* de impressão);
  - Ordenação do encaminhamento dos pacotes em um roteador;
  - *Buffer* em um *stream* de vídeo (*youtube*);



- Operações Básicas em uma lista:
  - Criação;
  - Destruição;
  - Inserção de um elemento (*push*);
  - Remoção de um elemento (*pop*);
  - Leitura do primeiro elemento da lista (*front*);

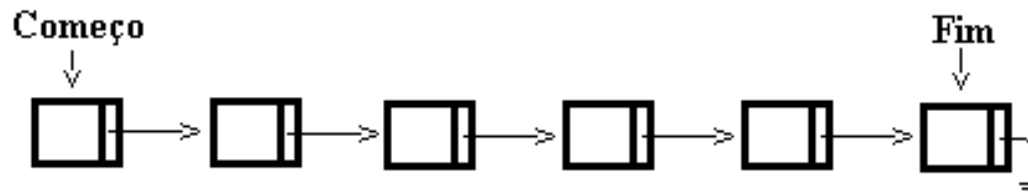
- **Fila Sequencial:**

- Faz uso de *arrays* para armazenar os dados;
- Possui tamanho fixo (alocação sequencial);
- Depois que a quantidade máxima de elementos é alcançada, só é permitido adicionar um novo elemento caso o primeiro seja removido;



- **Fila Encadeada:**

- Utiliza listas encadeadas;
- NÃO possui tamanho fixo (alocação dinâmica);
- Cada elemento (NÓ) possui um ponteiro para o próximo elemento da fila;



## Fila Sequencial

# FILA SEQUENCIAL

## → Implementação

- O primeiro passo para a criação de uma lista sequencial (estática) é definir a quantidade máxima de elementos ( $N$ ) e alocar memória necessária
- O processo de inserção e remoção fará com que a fila ande no vetor.



- Exemplo:
  - **Inserir** os elementos:
    - 1.4, 2.2, 3.5, 4.0
  - Em seguida **remove** dois elementos



# FILA SEQUENCIAL

## → Implementação

- Exemplo:
  - Inserir os elementos:
    - 1.4, 2.2, 3.5, 4.0
- **push(1.4)**



# FILA SEQUENCIAL

## → Implementação

- Exemplo:
  - Inserir os elementos:
    - 1.4, 2.2, 3.5, 4.0
- **push(2.2)**

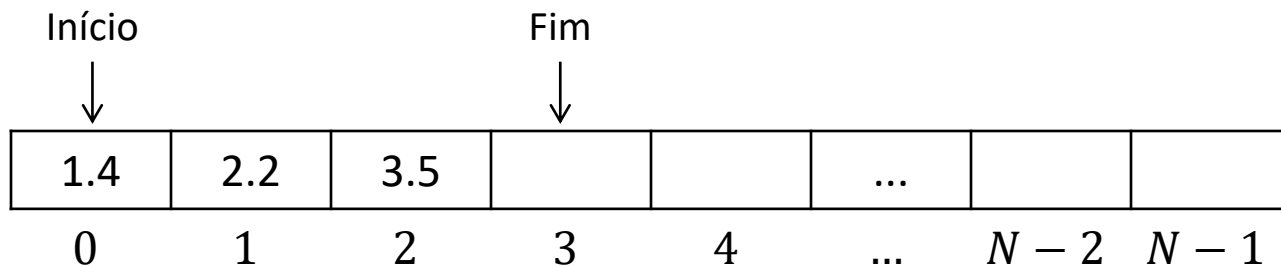




# FILA SEQUENCIAL

## → Implementação

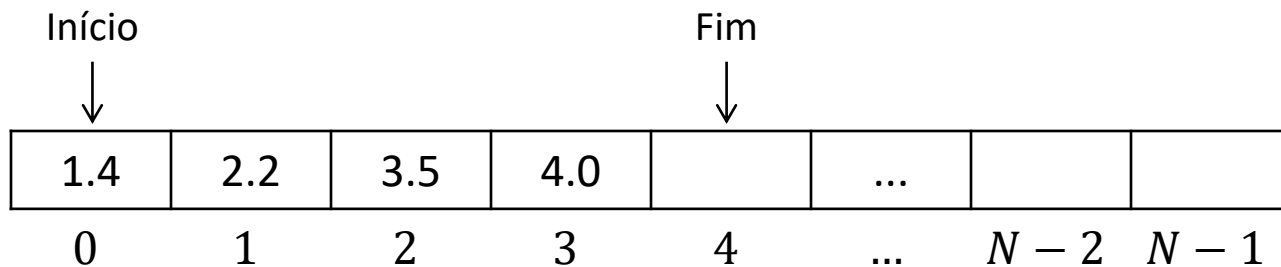
- Exemplo:
  - Inserir os elementos:
    - 1.4, 2.2, 3.5, 4.0
- **push(3.5)**



# FILA SEQUENCIAL

## → Implementação

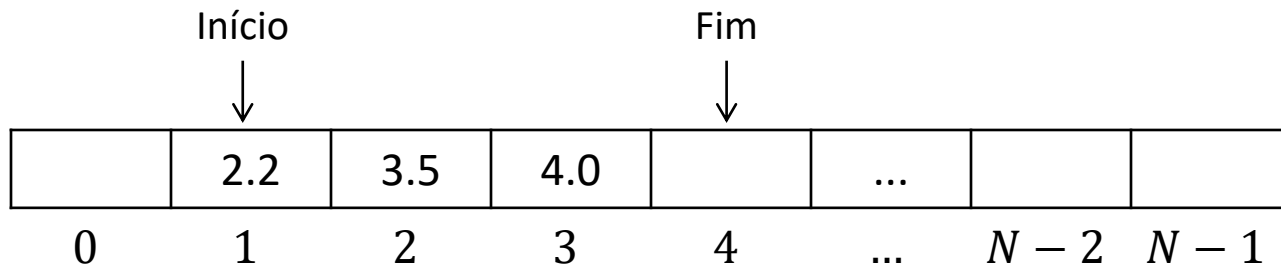
- Exemplo:
  - Inserir os elementos:
    - 1.4, 2.2, 3.5, 4.0
- **push(4.0)**



# FILA SEQUENCIAL

## → Implementação

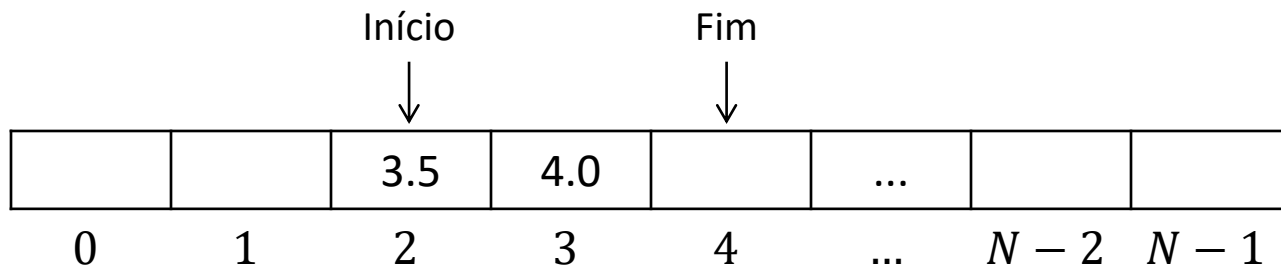
- Exemplo:
  - Remover dois elementos:
    - Realizar a operação `pop()` duas vezes.
- `pop()`



# FILA SEQUENCIAL

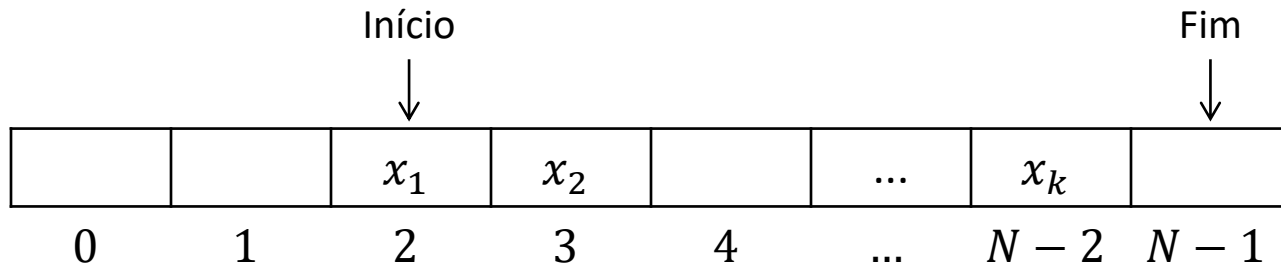
## → Implementação

- Exemplo:
  - Remover dois elementos:
    - Realizar a operação `pop()` duas vezes.
- `pop()`, `pop()`



## → Implementação

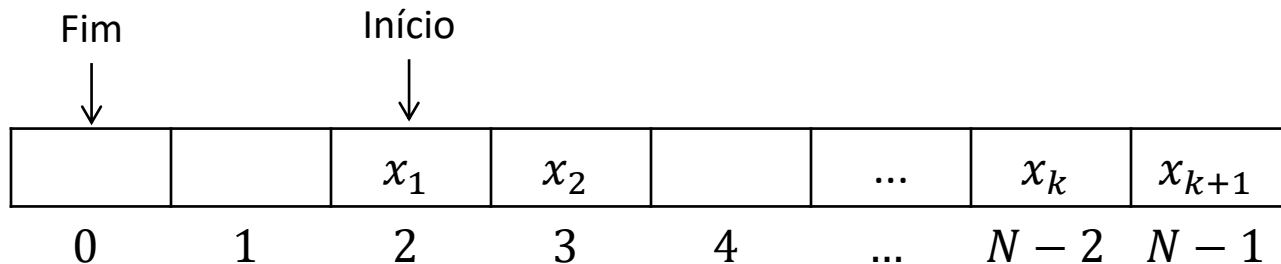
- Observe que em dado instante a parte ocupada do vetor pode chegar à ultima posição;
- Para reaproveitar as primeiras posições livres do vetor sem redistribuir os elementos podemos incrementar as posições do vetor de forma “circular”.



# FILA SEQUENCIAL

## → Implementação

- Observe que em dado instante a parte ocupada do vetor pode chegar à ultima posição;
- Para reaproveitar as primeiras posições livres do vetor sem redistribuir os elementos podemos incrementar as posições do vetor de forma “circular”.
- $\text{push}(x_{k+1})$



## → Implementação

- Observe que em dado instante a parte ocupada do vetor pode chegar à ultima posição;
- Para reaproveitar as primeiras posições livres do vetor sem redistribuir os elementos podemos incrementar as posições do vetor de forma “circular”.
- $\text{push}(x_{k+1})$ ,  $\text{push}(x_{k+2})$

